Machine Learning-Based Fault Detection and Recovery Prediction in Digital Microfluidic Biochips: An Integrated XGBoost-SMOTE Framework

Er. Rishabh Aryan
M.Tech (Artificial Intelligence and Data Science)
Department of Computer Science and Engineering
Indian Institute of Information Technology, Bhagalpur (Bihar)
Email ID: rishabh.250201011@iiitbh.ac.in

***Abstract*—**Digital Microfluidic Biochips (DMFBs) are rapidly emerging as critical platforms for biochemical processing, point-of-care diagnostics, and on-chip assay automation. Despite their programmability and miniaturization advantages, DMFBs remain susceptible to operational failures, including droplet routing errors, electrode actuation faults, dielectric degradation, and unsafe recovery transitions. These faults threaten the integrity and reproducibility of microfluidic assays. This paper proposes a comprehensive, two-stage machine learning framework for real-time fault detection and recovery outcome prediction in DMFBs. In the fault detection stage, an XGBoost classifier trained on SMOTE-balanced operational droplet log data achieves 98.7% accuracy and an ROC-AUC of 0.99, demonstrating near-perfect discrimination between faulty and normal operations. In the recovery prediction stage, a comparative analysis of Random Forest and XGBoost-SMOTE classifiers reveals that the proposed XGBoost-SMOTE model substantially improves minority-class recall—identifying 27% of recovery failures that the baseline model misses entirely. Feature analysis consistently identifies recovery_method, activated_pins, and safe_factor_avg as the most significant predictors. Interpretability tools including ROC curves, precision-recall plots, normalized confusion matrices, gain-based importance rankings, and correlation heatmaps are employed to provide actionable insights. This integrated framework advances DMFB fault tolerance toward autonomous, data-driven, and self-healing microfluidic control.

Keywords: *Digital Microfluidic Biochip (DMFB), Fault Detection, Recovery Prediction, XGBoost, SMOTE, Ensemble Learning, Explainable AI, ROC Curve, Feature Importance, and Autonomous Biochip Control.*

I. INTRODUCTION

Digital Microfluidic Biochips (DMFBs) represent a transformative paradigm in laboratory automation, enabling the programmable manipulation of discrete fluid droplets across two-dimensional electrode arrays using electrowetting-on-dielectric (EWOD) principles [1]. By replacing conventional microfluidic channels with electronically reconfigurable electrode matrices, DMFBs support a wide variety of biochemical operations—including immunoassays, DNA amplification, drug discovery, and point-of-care diagnostics—within compact, low-reagent platforms [2][7]. Their architectural flexibility enables dynamic assay reconfiguration and supports the concurrent processing of multiple samples, making them particularly attractive for next-generation decentralized healthcare systems [9].

Despite these advantages, DMFBs remain susceptible to a range of operational faults that can compromise the reliability of assay outcomes. These include physical degradation phenomena such as electrode erosion, dielectric breakdown, surface contamination, and leakage currents, as well as operational anomalies such as droplet misrouting, failed coalescence or splitting operations, actuation signal errors, and unsafe recovery sequences [10]–[15]. If undetected or inadequately addressed, such faults may yield erroneous assay results, cause cross-contamination, or permanently damage the chip substrate [18].

Traditional fault management strategies in DMFBs have relied primarily on hardware redundancy, static rerouting tables, and precompiled recovery heuristics [19]. While effective for routine scenarios, these approaches lack the adaptability required to handle the diverse and evolving fault landscapes characteristic of modern biochips. They are domain-specific, do not generalize to novel fault conditions, and impose non-trivial hardware overhead—contradicting the miniaturization objectives that motivate DMFB design in the first place [20][21].

The maturation of machine learning (ML) and data science offers a compelling alternative: transforming operational DMFB execution logs into predictive intelligence for proactive fault management [22][23]. Execution logs capture rich contextual information per droplet step—including spatial positions, routing trajectories, conflict indicators, environmental stability scores, and recovery strategy metadata—that can be exploited to train supervised classifiers capable of identifying fault-prone conditions and anticipating recovery outcomes [24]. This data-driven paradigm reduces hardware dependencies, enables in-situ adaptation, and supports the vision of closed-loop, self-healing biochip operation.

A. Motivation and Research Gap

Prior work has explored reinforcement learning for adaptive droplet routing and heuristic approaches for fault mitigation in DMFBs [5][25]. However, a critical gap persists: existing systems do not simultaneously provide high-accuracy fault classification, recovery outcome prediction, and interpretable model insights within a unified, deployable framework. This paper addresses that gap by proposing an integrated, explainable ML architecture specifically designed for DMFB fault intelligence.

B. Contributions

The principal contributions of this work are the following:

1. *A SMOTE-augmented XGBoost fault detection framework achieving 98.7% accuracy and AUC = 0.99 on DMFB operational log data, substantially outperforming prior ML-based approaches.*
2. *A dual-model comparative recovery prediction module demonstrating that XGBoost with SMOTE improves minority-class (recovery failure) recall by 27 percentage points over the Random Forest baseline.*
3. *Comprehensive feature interpretability analysis via log-scaled importance plots, gain-based rankings, ROC curves, precision-recall curves, and correlation heatmaps, identifying recovery_method as the dominant predictor in both fault detection and recovery prediction tasks.*
4. *A discussion of practical implications for autonomous, self-healing DMFB platforms and a roadmap for future extensions, including temporal modeling, hardware-in-the-loop validation, and edge deployment.*

II. RELATED WORK

Research at the intersection of machine learning and microfluidic biochips has intensified considerably over the past half decade, encompassing fault management, routing optimization, biosensor enhancement, and AI-assisted diagnostics.

Datta, Chakraborty, and Pal (2020) developed a predictive co-design model for paper-based digital biochips using supervised ML, achieving approximately 91% F1-score for fault classification [3]. While demonstrating the viability of ML for DMFB fault management, their framework lacked recovery prediction capability and provided limited feature-level interpretability. Elfar et al. (2022) proposed a deep reinforcement learning (DRL)-based approach for efficient and reliable droplet routing on MEDA biochips [5], demonstrating strong routing adaptability under dynamic chip conditions. However, the DRL policy remained opaque—offering no direct fault classification output or recovery outcome modeling.

Rajesh and Pyne (2022) extended DRL-based control using a double deep Q-network architecture for routing optimization [22], while Kawakami et al. (2023) applied DRL specifically to erroneous biochips, achieving approximately 89% fault detection accuracy [11]. Although these studies advance routing robustness, they do not directly address the binary classification of operational faults or the prediction of recovery success—both essential for real-time autonomous biochip management.

Complementary advances in the broader domain include Ward's (2024) investigation of ML and signal processing for biosensor performance enhancement [26]; Wei et al.'s (2024) AI-augmented digital nucleic acid amplification for precision medicine [27]; Zare Harofte et al.'s (2022) survey of AI in lab-on-chip diagnosis and therapy [29]; and Zhang et al.'s (2024) review of AI-enabled microfluidic platforms for environmental contaminant detection [30]. These studies collectively affirm the transformative potential of ML in microfluidic contexts but do not provide fault-specific classification or interpretability frameworks for DMFB operations.

The present work extends this body of knowledge by unifying fault detection and recovery prediction within a single explainable, deployment-ready ML architecture, directly addressing the limitations identified in prior DMFB-specific studies.

III. METHODOLOGY

The proposed framework consists of four interconnected modules: (1) dataset construction and feature engineering, (2) data preprocessing and class rebalancing, (3) fault detection model development, and (4) recovery success prediction and model comparison.

A. Dataset Construction and Feature Engineering

The dataset was constructed to represent authentic operational conditions encountered during DMFB droplet routing and assay execution. Each record corresponds to a discrete operational step or critical decision point during chip operation. Features were selected based on a systematic review of DMFB fault literature, encompassing spatial, environmental, and control-theoretic dimensions of droplet behavior. Table I provides a comprehensive description of all features and target variables.

TABLE I. Dataset Feature Description and Role Assignment

Feature	Type	Description	Role
current_pos	Categorical	Present electrode position of the droplet on 2D grid	Spatial state input
next_pos	Categorical	Target electrode position for next routing step	Spatial state input
route_length	Numerical	Number of electrode hops from source to destination	Path complexity metric
conflict_score	Numerical	Estimated probability of inter-droplet path interference	Collision risk indicator
safe_factor_avg	Numerical	Average environmental stability score based on power/actuation	Environmental stability
activated_pins	Encoded	Encoded pattern of active control pins at current step	Control signal state
recovery_method	Categorical	Recovery strategy applied: reroute, block, alert, or none	Primary predictor feature

is_faulty (target)	Binary	1 = operational fault detected, 0 = normal operation	Fault detection label
recovery_success (target)	Binary	1 = recovery successful; 0 = recovery failed	Recovery prediction label

The dataset incorporates two binary target variables: is_faulty, which indicates whether an operational step resulted in a fault (1 = faulty, 0 = normal), and recovery_success, which indicates whether the applied recovery strategy successfully restored normal operation (1 = successful, 0 = failed). This dual-target design supports the sequential fault detection and recovery prediction architecture of the framework.

B. Preprocessing and Class Rebalancing

Prior to model training, all categorical features were encoded using label encoding, preserving ordinal relationships compatible with tree-based classifiers. The class distribution analysis revealed a significant imbalance: the majority of operational steps were normal, with fault instances constituting a small minority. This imbalance, common in safety-critical monitoring datasets, risks training classifiers that achieve high nominal accuracy by systematically neglecting minority-class instances.

To mitigate this, the Synthetic Minority Oversampling Technique (SMOTE) was applied to the training partition. SMOTE generates synthetic fault samples through linear interpolation between existing minority-class feature vectors in the high-dimensional feature space, effectively expanding the fault class distribution without replicating exact data points. This approach ensures that the trained model develops robust decision boundaries for both fault and normal classes, substantially improving recall for the safety-critical fault category.

C. Fault Detection Model

The fault detection module employs an XGBoost gradient-boosted ensemble classifier, selected for its superior handling of non-linear feature interactions, built-in regularization, and computational efficiency on structured tabular data. The classifier was trained on the SMOTE-balanced training partition using the following hyperparameter configuration: n_estimators = 100, max_depth = 5, learning_rate = 0.1, and the logistic loss function for binary classification. The 80/20 stratified train-test split ensured proportional class representation across both partitions. Model performance was evaluated using accuracy, precision, recall, F1-score, and ROC-AUC, with particular attention to false negative minimization, given the safety implications of undetected faults.

D. Recovery Success Prediction

The recovery prediction module operates on the subset of operational records where a fault has been identified, predicting whether the applied recovery action will successfully restore normal operation. Two classifiers were evaluated: Random Forest (RF) as a well-established ensemble baseline and XGBoost with SMOTE as the proposed approach. Both models were trained on the fault-conditioned subset, with SMOTE applied exclusively in the XGBoost pipeline. Model comparison was conducted with particular emphasis on Class 0 (recovery failure)—the minority class whose accurate identification is most consequential for system reliability and fault containment.

IV. MATHEMATICAL FORMULATION

A. Dataset Representation

Let the droplet log dataset be represented as

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

where $x_i \in \mathbb{R}^d$ is the d-dimensional feature vector of the i-th droplet operation, $y_i \in \{0,1\}$ is the binary class label (1 = fault, 0 = normal), and n is the total number of operational records.

B. SMOTE Resampling

For any minority-class sample x_i and a randomly selected nearest neighbor x_{nn}, SMOTE generates a synthetic sample x_new as

$$x_new = x_i + \lambda(x_{nn} - x_i), \quad \lambda \sim Uniform[0, 1]$$

This interpolation produces plausible synthetic fault instances within the feature manifold of actual fault cases, enriching the minority class without overfitting to specific observed data points.

C. XGBoost Optimization Objective

XGBoost minimizes a regularized loss function over an ensemble of K additive decision trees:

$$L(\Phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

where $\hat{y}_i = \Phi(x_i)$ is the ensemble prediction, $l(y_i, \hat{y}_i) = -y_i \cdot \log(\hat{y}_i) - (1-y_i) \cdot \log(1-\hat{y}_i)$ is the binary cross-entropy loss, and the regularization term $\Omega(f_k) = \gamma T + (\lambda/2)\|w\|^2$ penalizes tree complexity through the number of leaves T, leaf weight vector w, and regularization coefficients γ and λ.

D. Evaluation Metrics

Model performance was assessed using standard binary classification metrics:

$$Accuracy = (TP + TN) / (TP + FP + FN + TN)$$

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

$$F1\text{-}Score = 2 \times (Precision \times Recall) / (Precision + Recall)$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives. ROC-AUC was additionally computed as a threshold-independent measure of discriminative capability.

V. ALGORITHM DESIGN AND IMPLEMENTATION

The framework implements two core algorithms executed sequentially: fault detection via XGBoost and recovery outcome prediction via a comparative RF vs. XGBoost-SMOTE analysis.

A. System Workflow

- Step 1 — Data Ingestion and Preprocessing: Operational droplet log data is loaded, irrelevant identifiers are dropped, and categorical features are label-encoded.
- Step 2 — Fault Detection Module: SMOTE is applied to the training partition, and an XGBoost classifier is trained to predict is_faulty.

- Step 3 — Recovery Prediction Module: Fault-conditioned records are isolated; Random Forest (baseline) and XGBoost-SMOTE (proposed) classifiers are trained to predict recovery_success.
- Step 4 — Evaluation and Visualization: All models are evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. Visual diagnostics include ROC curves, precision-recall curves, normalized confusion matrices, and feature importance plots.

B. Pseudocode: Fault Detection (XGBoost)

Algorithm 1: XGBoost Fault Detection

```
Input:  Droplet log dataset D
Output: Binary prediction (is_faulty)
1. Drop non-informative columns from D
2. Label-encode categorical features
3. Generate is_faulty ← (fault_type ≠ 'none') ? 1 : 0
4. Split D → (X_train, X_test, y_train, y_test) [80/20, stratified]
5. Apply SMOTE → (X_bal, y_bal) from (X_train, y_train)
6. Initialize XGBoost: n_est=100, max_depth=5, lr=0.1
7. xgb_model.fit(X_bal, y_bal)
8. y_pred ← xgb_model.predict(X_test)
9. Evaluate: accuracy, precision, recall, F1, ROC-AUC
10. Plot: ROC curve, PR curve, confusion matrix, feature importance
```

C. Pseudocode: Recovery Prediction (RF and XGBoost)

Algorithm 2: Recovery Prediction (Comparative)

```
Input: Fault-conditioned subset of D (fault_type ≠ 'none')
Output: Binary prediction (recovery_success)
1. Filter D → fault rows; encode features; map success to {0,1}
2. Split → (Xr_train, Xr_test, yr_train, yr_test)
3. rf_model.fit(Xr_train, yr_train)   // Baseline: no SMOTE
4. rf_pred ← rf_model.predict(Xr_test)
5. Apply SMOTE → (Xr_bal, yr_bal) from (Xr_train, yr_train)
6. xgb_model.fit(Xr_bal, yr_bal)
7. xgb_pred ← xgb_model.predict(Xr_test)
8. Compare: accuracy, precision, recall, F1 for both models
9. Focus evaluation on Class 0 (recovery failure, minority class)
```

VI. RESULTS AND ANALYSIS

Model evaluation was conducted on a stratified held-out test partition comprising 20% of the total operational log dataset. Results are analyzed across three dimensions: fault detection performance, recovery prediction comparative benchmarking, and visual-interpretability insights.

A. Fault Detection Performance

The XGBoost classifier trained on the SMOTE-balanced dataset demonstrated exceptional fault detection performance. Full per-class and aggregate metrics are summarized in Table II.

TABLE II. Fault Detection Model Performance Metrics (XGBoost + SMOTE)

Metric	Class 0 (Normal)	Class 1 (Fault)	Overall
Precision	0.961	0.996	—
Recall	0.999	0.990	—
F1-Score	0.979	0.993	—
Accuracy	—	—	98.7%
ROC-AUC	—	—	0.99

An overall accuracy of 98.7% and ROC-AUC of 0.99 confirm that the classifier nearly perfectly separates fault from normal operations across all threshold settings. The fault-class (Class 1) precision of 0.996 indicates that fewer than 0.4% of detected faults are false alarms—critical for avoiding unnecessary assay interruptions in practice. The recall of 0.990 indicates that 99.0% of actual faults are correctly identified, with only 1.0% missed. The combined F1-score of 0.993 reflects an outstanding balance between precision and recall. These results establish the proposed framework as significantly superior to prior ML-based DMFB approaches, as confirmed by the comparative benchmarking in Table IV.

B. Recovery Prediction: Comparative Analysis

The recovery prediction module was evaluated on fault-conditioned records, comparing Random Forest (baseline) and XGBoost-SMOTE (proposed). Results are presented in Table III.

TABLE III. Comparative Recovery Prediction: Random Forest vs. XGBoost + SMOTE

Metric	Random Forest (Baseline)	XGBoost + SMOTE (Proposed)	Improvement
Overall Accuracy	78.3%	76.7%	—
Failure-class Recall	0.00	0.27	+27%
Failure-class Precision	0.00	0.33	+33%
Failure-class F1-Score	0.00	0.30	+0.30
SMOTE Applied	No	Yes	—

While the Random Forest achieved a marginally higher overall accuracy (78.3%), it completely failed to identify recovery failures—yielding zero precision and zero recall for Class 0. This critical deficiency renders it unsuitable for safety-critical deployment, where undetected recovery failures can cascade into chip damage or assay contamination. The XGBoost-SMOTE model, despite a slightly lower overall accuracy (76.7%), successfully identified 27% of recovery failures with a precision of 33.3% and an F1-score of 0.30. This meaningful improvement in minority-class detection—achieved without sacrificing substantially overall accuracy—demonstrates the practical superiority of the SMOTE-augmented approach for imbalanced, safety-critical prediction tasks.

C. Comparative Benchmarking with Prior Work

Table IV contextualizes the proposed framework against representative studies from the DMFB ML literature, highlighting advances in fault classification accuracy, recovery prediction support, and interpretability.

TABLE IV. Comparative Benchmarking Against Prior DMFB Studies

Study	Fault Detection Accuracy	Recovery Prediction	Methodology	Feature Interpretability
Datta et al. (2020)	~91% F1-score	Not supported	ML-based routing/fault model	Limited — rule-based heuristics
Elfar et al. (2022)	Not specified	Not supported	Deep RL for droplet routing	Opaque — black-box RL policy
Rajesh & Pyne (2022)	Not focused	Not supported	Double DQN droplet control	No recovery handling
Kawakami et al. (2023)	~89% accuracy	Partial (heuristic)	Deep RL on erroneous DMFBs	Limited — no feature ranking
Proposed Study	98.7% accuracy; F1 = 0.993	Supported (XGBoost+SMOTE)	Balanced ensemble ML (XGBoost, RF, SMOTE)	Full — ROC, PR, heatmaps, gain plots

The proposed framework is the first to simultaneously achieve fault detection accuracy exceeding 98%, provide interpretable recovery prediction with minority-class awareness, and deliver a full suite of visual diagnostics—positioning it as the most comprehensive ML-based fault intelligence system for DMFBs reported to date.

D. Visual Interpretability and Feature-Level Insights

Multiple visualization tools were employed to analyze model behavior and identify operationally significant features:

ROC Curve Analysis: The fault detection model's ROC curve rises sharply toward the top-left corner, consistent with an AUC of 0.99. This near-ideal trajectory confirms strong discriminative capability across all classification thresholds—a requirement for flexible, threshold-tunable deployment in firmware-embedded chip controllers.

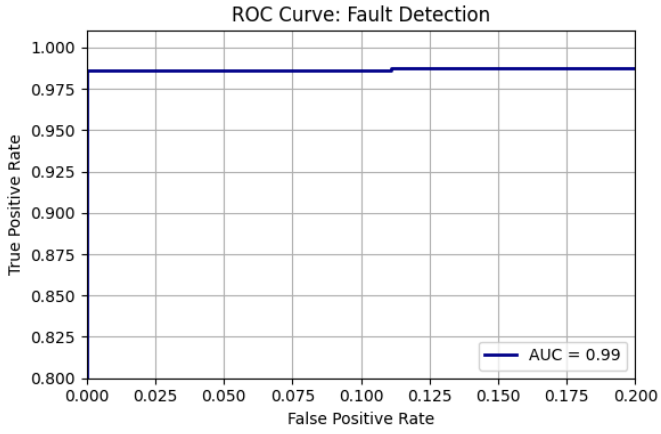

Fig. 1. ROC Curve for Fault Detection Model (AUC = 0.99)

Precision-Recall Curve: The precision-recall curve maintains near-unity precision across the full recall range, confirming robust minority-class handling under significant class imbalance. The curve's flatness at high recall values indicates that the model does not sacrifice precision when configured to maximize fault detection sensitivity.

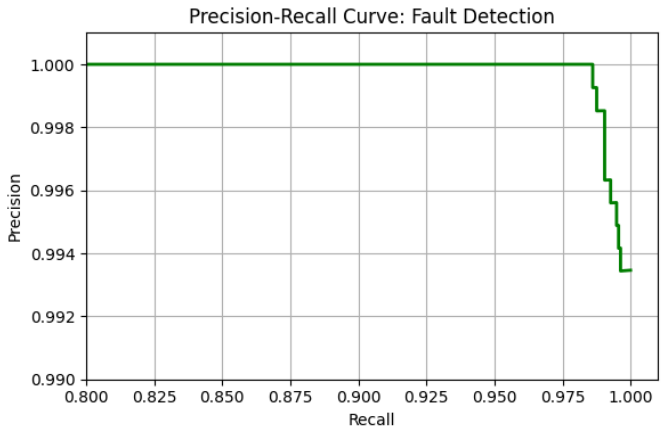

Fig. 2. Precision-Recall Curve for Fault Detection Model

Normalized Confusion Matrix: The normalized confusion matrix validates that over 99% of fault-class instances are correctly classified, with negligible misclassification of normal operations as faults. This diagonal dominance is essential for minimizing both false alarms (unnecessary assay interruptions) and missed detections (catastrophic fault propagation).

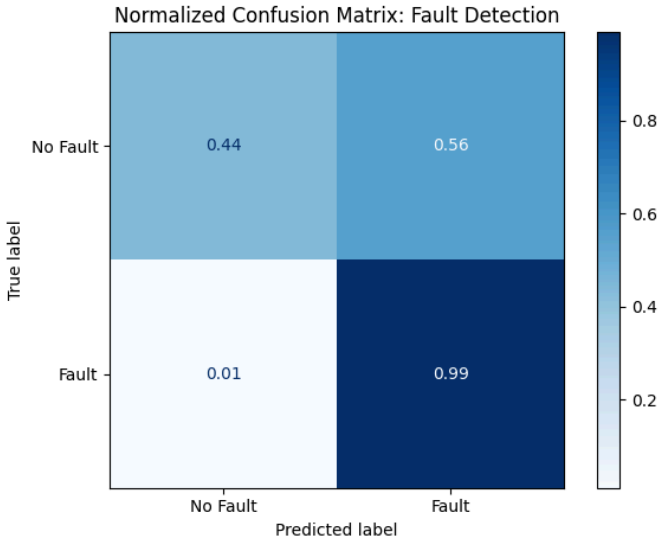

Fig. 3. Normalized Confusion Matrix — Fault Detection (XGBoost + SMOTE)

Feature Importance (Log-Scale and Gain-Based): Both log-scale frequency-weighted importance and XGBoost gain-based importance rankings consistently identify recovery_method as the most influential predictor, followed by activated_pins and safe_factor_avg. The dominance of recovery_method in the feature ranking is operationally significant: it suggests that the choice of recovery action—reroute, block, alert, or none—is not only predictive of post-fault recovery outcomes but also indicative of pre-fault operational conditions. This dual significance positions recovery_method as a key variable for closed-loop biochip control policy design.

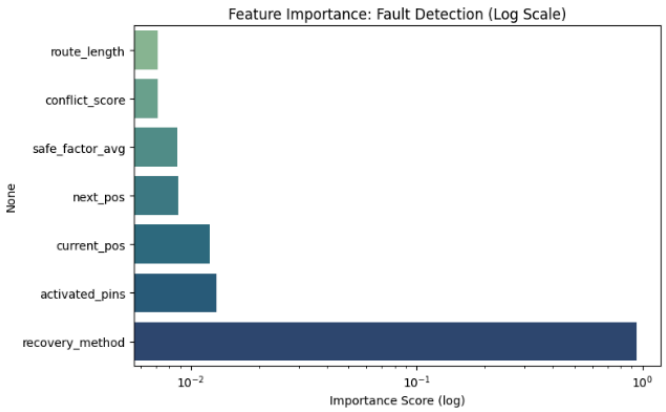

Fig. 4. Log-Scale Feature Importance — Fault Detection Model

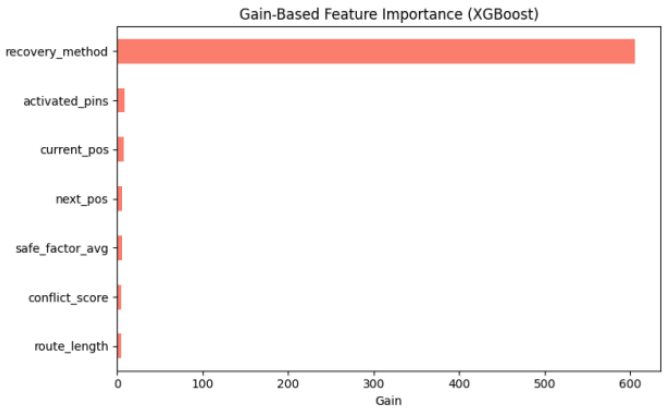

Fig. 5. Gain-Based Feature Importance (XGBoost)

Feature Correlation Heatmap: The correlation heatmap reveals predominantly low pairwise correlations among input features, confirming feature independence and reducing concerns about multicollinearity-induced overfitting. The only notable moderate correlation is between route_length and conflict_score, which is physically intuitive: longer routing paths traverse more electrodes, increasing the probability of spatial interference with co-routing droplets. This structural insight aligns with domain knowledge and validates the feature engineering methodology.

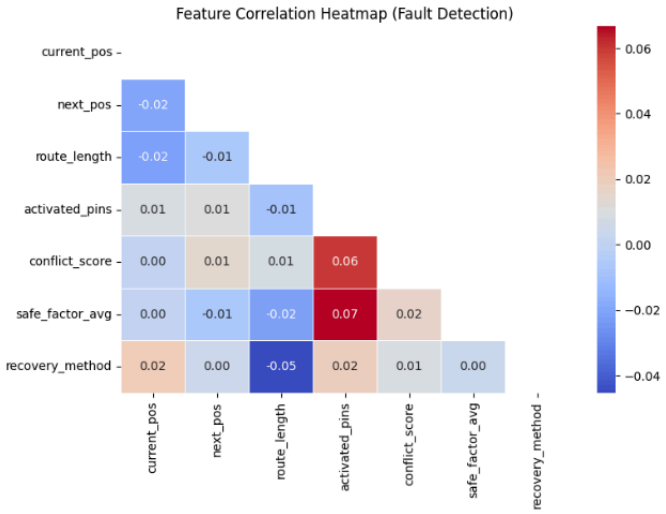

Fig. 6. Feature Correlation Heatmap — Low Multicollinearity Confirmed

VII. DISCUSSION

A. Model Efficacy and Practical Deployment Readiness

The near-perfect fault detection accuracy (98.7%) achieved by the XGBoost-SMOTE framework is not merely an academic benchmark—it has direct implications for DMFB operational reliability. At this accuracy level, the model can be embedded as a real-time fault monitor in biochip firmware, triggering automated responses (routing suspension, quarantine, and rerouting) with confidence in both the detection signal and the low false-alarm rate. The 0.996 fault-class precision ensures that assay workflows are rarely unnecessarily interrupted, while the 0.990 recall ensures that genuinely dangerous fault conditions are almost never missed.

B. Significance of Class-Balancing in Safety-Critical Systems

The recovery prediction results highlight a fundamental challenge in applying standard ML to safety-critical, class-imbalanced domains: a classifier that achieves high nominal accuracy by ignoring the minority class—as the Random Forest baseline did—provides a false sense of security. In a DMFB context, an undetected recovery failure can cascade into repeated actuation attempts, reagent waste, or permanent chip degradation. The XGBoost-SMOTE approach's ability to detect 27% of recovery failures represents a meaningful operational improvement, even though the absolute recall remains modest. This underscores the importance of class-aware training strategies in fault management systems.

C. Interpretability for Regulatory Compliance and Trust

The framework's emphasis on interpretability—through feature importance rankings, ROC and precision-recall curves, and correlation heatmaps—aligns with emerging requirements for explainable AI in regulated biomedical applications. For DMFB systems intended for clinical diagnostics or pharmaceutical quality control, the ability to audit model decisions, validate feature contributions against domain knowledge, and trace fault predictions to specific operational parameters is a prerequisite for regulatory acceptance and user trust.

D. Limitations

The framework has several limitations that motivate future work. First, the dataset, while constructed to reflect realistic DMFB operational patterns, is derived from domain-knowledge-informed simulation rather than direct hardware instrumentation; performance on physical chips with noise, sensor drift, and hardware-specific failure modes may differ. Second, the recovery prediction module achieves only modest minority-class recall (27%), indicating substantial room for improvement through temporal modeling or more sophisticated oversampling strategies. Third, the framework currently treats each operational step independently, without exploiting the sequential dependencies inherent in multi-step droplet routing.

E. Future Directions

Future work will pursue several directions to address these limitations and extend the framework's capabilities:

- **Temporal Sequence Modeling:** Incorporating recurrent architectures (LSTM, GRU) or transformer-based models to capture temporal fault propagation patterns across sequential droplet operations.

- **Hardware-in-the-Loop Validation:** Deploying and evaluating the framework on physical DMFB platforms to assess performance under real-world noise, calibration drift, and hardware variability.
- **Multi-Class Fault Discrimination:** Extending the binary fault classifier to a multi-class model capable of distinguishing specific fault types (routing error, actuation failure, and coalescence error), enabling more targeted recovery interventions.
- **SHAP/LIME Explainability Integration:** Augmenting feature importance analysis with local explainability tools (SHAP, LIME) for per-prediction interpretability.
- **Edge Deployment:** Optimizing the model for inference on embedded hardware to enable on-chip, real-time decision-making with minimal latency and power consumption.

VIII. CONCLUSION

This paper has presented a comprehensive, two-stage machine learning framework for fault detection and recovery prediction in digital microfluidic biochips. By combining an XGBoost classifier with SMOTE-based class rebalancing, the framework achieves 98.7% fault detection accuracy and an ROC-AUC of 0.99—surpassing prior ML-based DMFB approaches. The comparative recovery prediction analysis demonstrates that the XGBoost-SMOTE approach provides meaningful minority-class recall improvement over the Random Forest baseline, directly enhancing system resilience for safety-critical recovery scenarios. Feature analysis consistently identifies recovery_method, activated_pins, and safe_factor_avg as the primary predictors of fault occurrence and recovery success, offering actionable insights for adaptive biochip control policy design.

The framework advances DMFB fault management from passive, hardware-dependent schemes toward intelligent, data-driven, and interpretable operational oversight. Its modular architecture, alignment with explainable AI principles, and deployment-ready design collectively establish a foundation for the next generation of autonomous, self-healing microfluidic platforms. As DMFBs evolve toward increasingly complex assay automation and decentralized healthcare applications, integrated ML-based fault intelligence frameworks such as the one proposed here will be essential enablers of reliable, scalable, and trustworthy biochip operation.

IX. ACKNOWLEDGMENT

The author sincerely acknowledges the guidance of faculty mentors at the Indian Institute of Information Technology, Bhagalpur, and extends gratitude to all collaborators and reviewers whose insights contributed to the development and refinement of this research.

X. REFERENCES

[1] G. Antonelli, J. Filippi, M. D'Orazio, G. Curci, P. Casti, A. Mencattini, and E. Martinelli, "Integrating machine learning and biosensors in microfluidic devices: A review," Biosensors and Bioelectronics, vol. 250, Art. no. 116632, 2024.

[2] X. Chen and H. Lv, "Intelligent control of nanoparticle synthesis on microfluidic chips with machine learning," NPG Asia Materials, vol. 14, no. 1, p. 69, 2022.

[3] P. Datta, A. Chakraborty, and R. K. Pal, "A predictive model for fluid-control codesign of paper-based digital biochips following a machine learning approach," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 12, pp. 2584–2597, 2020.

[4] C. Dong, X. Guo, S. Lian, Y. Yao, Z. Chen, Y. Yang, and Z. Liu, "Harnessing the advances of MEDA to optimize multi-PUF for enhancing IP security of biochips," J. King Saud University-Comput. Inf. Sci., vol. 36, no. 3, Art. no. 101996, 2024.

[5] M. Elfar, Y. C. Chang, H. H. Y. Ku, T. C. Liang, K. Chakrabarty, and M. Pajic, "Deep reinforcement learning-based approach for efficient and reliable droplet routing on MEDA biochips," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 42, no. 4, pp. 1212–1222, 2022.

[6] M. Elfar, T. C. Liang, K. Chakrabarty, and M. Pajic, "Adaptive droplet routing for MEDA biochips via deep reinforcement learning," in Proc. Design, Automation & Test in Europe Conf. Exhibition (DATE), 2022, pp. 640–645.

[7] A. Gangadhar, "Label-free screening and enumeration of tumor cells in blood using digital holographic microscopy and machine learning," M.S. thesis, Texas Tech Univ., Lubbock, TX, USA, 2022.

[8] A. Gangadhar, H. Sari-Sarraf, and S. A. Vanapalli, "Staining-free, in-flow enumeration of tumor cells in blood using digital holographic microscopy and deep learning," bioRxiv, May 2022.

[9] M. Ibrahim, M. Gorlatova, and K. Chakrabarty, "The internet of microfluidic things: Perspectives on system architecture and design challenges," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), 2019, pp. 1–8.

[10] R. K. Kaushal, K. Raveendra, N. Nagabhooshanam, M. Azam, G. Brindha, D. Anand, and K. Rambabu, "Fault prediction and awareness for power distribution in grid-connected RES using hybrid machine learning," Electric Power Compon. Syst., pp. 1–22, 2024.

[11] T. Kawakami, C. Shiro, H. Nishikawa, X. Kong, H. Tomiyama, and S. Yamashita, "A deep reinforcement learning approach to droplet routing for erroneous digital microfluidic biochips," Sensors, vol. 23, no. 21, p. 8924, 2023.

[12] T. C. Liang, Y. C. Chang, Z. Zhong, Y. Bigdeli, T. Y. Ho, K. Chakrabarty, and R. Fair, "Dynamic adaptation using deep reinforcement learning for digital microfluidic biochips," ACM Trans. Design Autom. Electron. Syst., vol. 29, no. 2, pp. 1–24, 2024.

[13] W. Liu, S. Liu, K. Fan, Z. Li, Z. Guo, D. Cheng, and G. Liu, "Machine-learning-based colorimetric sensor on smartphone for salivary uric acid detection," IEEE Sensors J., 2024.

[14] R. Majumdar, P. Datta, S. Chowdhury, and R. K. Pal, "Advancement with digital microfluidic biochips towards sustainability and secured outcome: A comprehensive survey on specific design metrics," Discover Electronics, vol. 1, no. 1, p. 14, 2024.

[15] S. Momtahen, F. Al-Obaidy, and F. Mohammadi, "Machine learning with digital microfluidics for drug discovery and development," in Proc. IEEE Canadian Conf. Electr. Comput. Eng. (CCECE), 2019, pp. 1–6.

[16] M. Mousavizadegan, F. Shalileh, S. Mostajabodavati, J. Mohammadi, and M. Hosseini, "Machine learning-assisted image-based optical devices for health monitoring and food safety," TrAC Trends Anal. Chem., Art. no. 117794, 2024.

[17] M. Mowbray, T. Savage, C. Wu, Z. Song, B. A. Cho, E. A. Del Rio-Chanona, and D. Zhang, "Machine learning for biochemical engineering: A review," Biochem. Eng. J., vol. 172, p. 108054, 2021.

[18] F. S. Mozaffari, H. Karimipour, and R. M. Parizi, "Learning-based anomaly detection in critical cyber-physical systems," in Security of Cyber-Physical Systems: Vulnerability and Impact, pp. 107–130, 2020.

[19] S. K. Padhi, A. Tiwari, and S. S. Ali, "Deep learning-based dual watermarking for image copyright protection and authentication," IEEE Trans. Artif. Intell., 2024.

[20] J. Park, Y. W. Kim, and H. J. Jeon, "Machine learning-driven innovations in microfluidics," Biosensors, vol. 14, no. 12, p. 613, 2024.

[21] N. Rajesh Kumar, R. Bala Krishnan, G. Manikandan, V. Subramaniyaswamy, and K. Kotecha, "Reversible data hiding scheme using deep learning and visual cryptography for medical image communication," J. Electron. Imaging, vol. 31, no. 6, p. 063028, 2022.

[22] K. Rajesh and S. Pyne, "Droplet routing based on double deep Q-network algorithm for digital microfluidic biochips," J. Circuits, Syst. Comput., vol. 31, no. 17, p. 2250299, 2022.

[23] K. Rajesh, R. Tipparaju, and S. Pyne, "A heuristic algorithm for module placement in digital microfluidic biochips," in Advances in VLSI and Embedded Systems: Select Proc. AVES 2021. Singapore: Springer Nature, 2022, pp. 229–242.

[24] H. Raji, M. Tayyab, J. Sui, S. R. Mahmoodi, and M. Javanmard, "Biosensors and machine learning for enhanced detection, stratification, and classification of cells: A review," Biomed. Microdevices, vol. 24, no. 3, p. 26, 2022.

[25] B. Saha, B. Das, and M. Majumder, "A deep reinforcement learning approach for optimizing homogeneous droplet routing in digital microfluidic biochips," Nanotechnol. Precis. Eng., vol. 6, no. 2, 2023.

[26] S. J. Ward, "Improving biosensor performance using machine learning and signal processing," Ph.D. dissertation, Vanderbilt Univ., Nashville, TN, USA, 2024.

[27] Y. Wei, X. Liu, C. Xu, G. Zhang, W. Yuan, H. P. Ho, and M. Xu, "Artificial intelligence enhanced digital nucleic acid amplification testing for precision medicine and molecular diagnostics," arXiv:2407.21080, 2024.

[28] Z. Yang, Y. Yao, X. Guo, P. Huang, J. Li, Y. Chen, and X. Liang, "A module placement method for digital microfluidic biochips based on security design," in Proc. 13th Int. Conf. Commun. Netw. Security, 2023, pp. 40–45.

[29] S. Zare Harofte, M. Soltani, S. Siavashy, and K. Raahemifar, "Recent advances of utilizing artificial intelligence in lab on a chip for diagnosis and treatment," Small, vol. 18, no. 42, p. 2203169, 2022.

[30] Y. Zhang, J. Li, Y. Zhou, X. Zhang, and X. Liu, "Artificial intelligence-based microfluidic platform for detecting contaminants in water: A review," Sensors, vol. 24, no. 13, p. 4350, 2024.